A COMPARATIVE ANALYSIS OF EVIDENTIAL REASONING AND CUMULATIVE SCORING ALGORITHMS IN THE CONTEXT OF A COMBAT IDENTIFICATION APPLICATION

June 2000

A. Mahalanabis, R. N. Lobbia and C. R. Willman
Boeing Military Aircraft & Missile Systems Group
The Boeing Company, P.O. Box 3707, MS 43-14, Seattle, WA 98124

ABSTRACT

Within the discipline of Combat ID, as related to high performance fighters, a problem of significant proportion that arises is how to determine quickly and accurately a target's ID. Various algorithms, many of which are proprietary, that address this problem have been developed over the last decade. But their robustness in handling a new and unforeseen target threat environment leaves much to be desired. In this paper we explore two candidate classes of algorithms for achieving this objective, namely, an evidential reasoning and a cumulative scoring technique. We present the results via a comparative analysis where we look at the problem not only from a timeliness and accuracy of identification perspective, but also from the point of view of computational throughput. The analysis is based on simulation results using the sensor fusion system of a high performance fighter program. This system accomplishes the fusion of attribute data from a diverse set of sensors in a real time, computationally constrained processing environment. We will show some of the performance advantages evidential reasoning exhibits over a cumulative scoring approach. This is demonstrated on a typical scenario that the fighter in question has to show performance against. Furthermore, we will discuss a methodology for using a by-product of the evidential reasoning algorithm as a score to help in the data association task of assigning sensor reports to system tracks.

1.0 INTRODUCTION

Often a major obstacle in developing a coherent strategy for fusion of attribute data on a high performance fighter program is the bewildering assortment of disparate sensors that provide attribute information. There are several well-known strategies for attribute fusion, which are candidate algorithms for such a task. In this paper we provide a comparative analysis of the performance of one such algorithm (the Dempster-Shafer method) and a variant of another well established paradigm (an ad-hoc scoring method). The analysis shows the performance of these algorithms on simulation data generated for a typical high performance fighter.

We do not, by any means, claim the superiority of one algorithm over the other. Our goal, instead, is to discuss the application of these two algorithms on a real life program. In this context, we find that both algorithms presented have their strengths and weaknesses. Although both algorithms seem to give about the same level of performance with regard to "time to convergence" to unambiguous ID, the scoring method is somewhat less computationally intensive. This however, is not a ringing endorsement of the scoring method. The analysis presented in this paper remains preliminary in nature. More robust testing of the two methods, under more strenuous scenario conditions must be undertaken before we can claim that one algorithm is better than the other. This work is left for a future date. However, we can make the claim that in relatively sparse scenario environments, the scoring method may provide an alternative to the more resource-expensive Dempster-Shafer technique.

The first step in understanding the problem one faces in the task of attribute fusion is to describe the type of attribute information we must fuse together. We begin, thus, by describing a typical sensor suite, and the type of

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (FROM - TO)
01-06-2000	Conference Proceedings	xx-xx-2000 to xx-xx-2000

4. TITLE AND SUBTITLE
A Comparative Analysis of Evidential Reasoning and Cumulative Scoring Algorithms in the Context of a Combat Identification Application
Unclassified

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Mahalanabis, A. ;
Lobbia, R. N. ;
Willman, C. R. ;

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME AND ADDRESS
Boeing Military Aircraft & Missile Systems Group
The Boeing Company
P.O. Box 3707, MS 43-14
Seattle, WA98124

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS
Director, CECOM RDEC
Night Vision and Electronic Sensors Directorate Security Team
10221 Burbeck Road
Ft. Belvoir, VA22060-5806

10. SPONSOR/MONITOR'S ACRONYM(S)
11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APUBLIC RELEASE
,

13. SUPPLEMENTARY NOTES
See Also ADM201258, 2000 MSS Proceedings on CD-ROM, January 2001.

14. ABSTRACT
Within the discipline of Combat ID, as related to high performance fighters, a problem of significant proportion that arises is how to determine quickly and accurately a target?s ID. Various algorithms, many of which are proprietary, that address this problem have been developed over the last decade. But their robustness in handling a new and unforeseen target threat environment leaves much to be desired. In this paper we explore two candidate classes of algorithms for achieving this objective, namely, an evidential reasoning and a cumulative scoring technique. We present the results via a comparative analysis where we look at the problem not only from a timeliness and accuracy of identification perspective, but also from the point of view of computational throughput. The analysis is based on simulation results using the sensor fusion system of a high performance fighter program. This system accomplishes the fusion of attribute data from a diverse set of sensors in a real time, computationally constrained processing environment. We will show some of the performance advantages evidential reasoning exhibits over a cumulative scoring approach. This is demonstrated on a typical scenario that the fighter in question has to show performance against. Furthermore, we will discuss a methodology for using a by-product of the evidential reasoning algorithm as a score to help in the data association task of assigning sensor reports to system tracks.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	Public Release	18	Fenster, Lynn lfenster@dtic.mil
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 703767-9007 DSN 427-9007

attribute information we receive from this sensor suite. We next provide a brief overview of the Dempster-Shafer method, and the ad-hoc Scoring Method. The scoring method we employ departs significantly from the conventional scoring techniques that are described in the literature. We describe in detail, the modifications we have made to the theory. A brief description of the testing philosophy we employed in performing the analysis of the simulation data follows. Interspersed with this discussion is a description of the specific scenario we utilized in our work. We provide detailed analysis results, and discuss the implications of these results for our particular application.

2.0 A DESCRIPTION OF THE SENSOR SUITE

A typical high performance fighter is equipped with a variety of sensor types. In continuing with our discussion, it becomes somewhat necessary to discuss what a typical sensor suite for such a fighter might entail. Broadly speaking, we are concerned with five distinct categories of sensors. These are Electronic Warfare (EW), Radar Measurements, Identification Friend or Foe (IFF), Off-board sources and Contextual Data.

The EW subsystem may use a variety of different functions to return attribute data. This data is typically in the form of a list of candidate platforms and associated confidences (say between 0 and 100). Similarly, IFF reports provide target identification information. Unlike the EW attribute data however, IFF cannot provide confidences for each of the candidate platforms it detects. Radar can employ Non-Cooperative Target Recognition (NCTR) to gain attribute information on observed targets. Again, such attribute data may take the form of platforms and confidences.

Our theoretical fighter can, in addition, rely on different sources of off-board attribute information. For example, it may receive track level ID information from other friendly fighters via a Fighter Data Link (FDL). The information broadcast across this data link may provide kinematic position data on other friendly fighters, or it may provide kinematic and attribute data on targets being tracked by other friendly fighters. Another source for off-board information may be a larger communication net, such as the Joint Tactical Information Distribution System (JTIDS). Assuming the fighter participates in this network, it would receive track level attribute information on friendly entities in the form of a data structure known as Precise Participant Location Information (PPLI). Both the FDL and JTIDS PPLI are distinguished from the other sensors in the suite by the fact that they provide track level information. As may be easily imagined, there are resulting implications for the fusion of such attribute information. Of primary concern is that by repeatedly fusing such information, we are essentially multi-counting the same information. Although algorithms have been developed based on the fusion methods we present in this paper to deal with this multi-counting issue, such a discussion is outside of the scope of this paper.

Most of the sensor systems we have discussed so far provide kinematic, as well as attribute information. Hence, every track has an associated kinematic estimate. See, for example, [1] for a description of how such a kinematic estimate may be derived. The maximum altitude and velocity that a variety of platforms can achieve is available in the form of mission data. These maximum values can be used to mask out those platforms that cannot possibly represent the target associated with a given track. That is, if track T has velocity v_t, then platform p, which has a maximum velocity $v_p < v_t$, clearly cannot be a candidate platform for track T. Although, strictly speaking, such an inference does not represent sensor data in the conventional sense, we list it here as a source of attribute information, and refer to it as contextual data. The mechanism for masking out non-viable candidates makes use of algorithms similar to those presented later in this paper.

Fundamentally, we note that regardless of the sensor subsystem, platform lists take one of two forms; they are either ambiguous lists of platforms, each of which has associated with it a confidence between 0 and 100, or they are ambiguous lists free of any associated confidence. We can generalize this even further by assuming that when no confidences are provided, every platform in a platform list has the same confidence. Hence, based on the sensor suite we have introduced in this section, we can assume that the type of evidence we are concerned with is of the form $\{(p_i, c_i) \mid p_i \in \mathbf{P}, 0 \le c_i \le 100\}$, where $\mathbf{P}$ is the set of all platforms in the universe of discourse.

3.0 THE CANDIDATE ALGORITHMS

In this section we provide a theoretical description of the two algorithms we are analyzing in this paper. We begin by briefly reviewing the Dempster-Shafer method of fusion. Assuming that all evidence to be fused is of the form described in the previous section, our first task is to derive a belief function from any given piece of evidence. Given a belief function, fusion of evidence proceeds according to Dempster's Rule of Combination. We describe the process of normalization and the pitfalls associated with such an operation. Finally, we describe the process of deriving a metric that measures the correctness of the correlation of a report with a track. This metric is a by-product of the Dempster-Shafer calculus.

We then proceed to a theoretical description of the scoring method. An additive process is employed in the fusion of evidence under this method. Associated with this additive process is the concept of the knowledge point, a theoretical point at which we can assume that enough evidence has been accumulated to begin the process of convergence to a single unambiguous platform. Convergence to a less ambiguous platform list is achieved via the process of pruning. Finally, although there is no clear by-product of the fusion process under this algorithm that can act as a measure of the correctness of correlation, we can derive such a metric. A description of a proposed metric is provided.

3.1 THE DEMPSTER-SHAFER CALCULUS

The precepts of the Dempster-Shafer Calculus (also referred to as Evidential Reasoning) are fairly well established in the domain of ID fusion. Here, we will give a cursory overview of the method.

As a preliminary to our discussion, let us fix the types of objects that we can observe in our universe. Let us refer to this set of objects as the *frame of discernment*. The frame of discernment is denoted by $\theta = \{x_1, x_2, \ldots, x_n\}$, where the various x_i are labels for the types of objects perceptible in the universe. In our case, the perceptible objects are friend and foe platforms of various types.

Let us say we are given some sort of a proposition *P*, in this universe. It is our task to determine the veracity of the proposition (In our specific context, the proposition we are most interested in say P_i, is of the form "Is the true ID of the observed target platform p_i?"). Let us say, based on some sort of a priori knowledge database, we have determined that the possibility that P_i is true is given by some number *y*. Let us further restrict the domain of any such *y* to the real valued unit interval $[0,1]$. Let us, therefore, denote the possibility of P_i being true as $p(P_i) = y$. How we derive the possibility for a given proposition is a function of the environment in which we are trying to implement Dempster-Shafer fusion. In the hypothetical environment we have constructed, the derivation of possibility is a straightforward matter. First, let us assume that with each sensor we have associated some trust factor t_i, such that $0 \le t_i \le 1$. This weight provides a measure of how much we trust the attribute data returned by the sensor. If we trust the data from a sensor absolutely, the trust factor associated with the sensor will be 1.0. If we half trust the data, the trust factor is 0.5 (and so on). Given a platform ID that consists of platform confidence pairs, derived from a sensor with trust factor t_i, we declare the possibility of any given platform $p(P_j)$, to be the product $t_i c_j$, where c_j is the confidence placed on the platform P_j.

It is important to emphasize that a possibility is not a probability. The summation of possibilities over an ambiguous set is not necessarily 1.0. In fact, in the majority of instances it is not 1.0. The possibility of an object being of type *P* is independent of the possibility of the same object being of type *Q*. In the Dempster-Shafer theory, the notion of a belief function is approximately analogous to the probability density function in probability theory. The difference between the two concepts lies in the notion of the assignment of belief mass (analogous to probability) to a set of object types, rather than to a single object type. In many situations, we may be able to say with some certainty that the type of an observed object is to be found in some set $\{P_1, P_2, \cdots, P_r\}$. But we may not be able to say with as much certainty that the object type is in the set $\{P_2, P_5, P_r\}$, for example, or in the set $\{P_i\}$. And so, we assign a larger amount of belief mass to the first set than the other two sets. A set to which we assign

belief mass is referred to as a focal element. Given focal elements $F_1, \cdots, F_m$, a belief function is a set of mass assignments

$$\begin{aligned} m(F_1) &= m_1 \\ m(F_2) &= m_2 \\ &\vdots \\ m(F_m) &= m_m \end{aligned}$$

Such that $m_1 + \cdots + m_m = 1.0$. Note that unlike possibilities, the masses on the focal elements of a belief function sum to 1.0. The question that we tackle next is given a possibility assignment arising from an ambiguous set, how do we create a belief function? In the next section, we describe a method of construction that is relatively straightforward and easy to understand.

3.1.2 CONSONANT BELIEF FUNCTION CONSTRUCTION

Given a set of platform possibility pairs, say $\{(P_1, p_1), \cdots, (P_r, p_r)\}$ how do we construct a belief function? To begin with, let us assume, without loss of generality, that $p_1 \geq p_2 \geq \cdots \geq p_r$. First of all, the domain of possibility is defined as $[0,1]$, as is the domain of belief. $P_1, \cdots, P_r$ can be thought of as being at least p_r possible. Certainly $P_1, \cdots, P_{r-1}$ may be more than p_r possible. But they are at least p_r possible. So, we write our belief in the focal element $P_1, \cdots, P_r$ is p_r, or $m(\{P_1, \cdots, P_r\}) = p_r$ (Here, p_r is called the *belief mass* on the focal element $P_1, \cdots, P_r$). Similarly, $P_1, \cdots, P_{r-1}$ are at least p_{r-1} possible. But note that p_r of this total possibility of p_{r-1} has already been accounted for in the last assignment of belief. So, we write $m(\{P_1, \cdots, P_{r-1}\}) = p_{r-1} - p_r$. Continuing on, we say $P_1, \cdots, P_{r-2}$ are at least p_{r-2} possible, but p_{r-1} of this possibility has already been accounted for. So we write $m(\{P_1, \cdots, P_{r-2}\}) = p_{r-2} - p_{r-1}$. If we continue this process, we get the following set of assignments.

$$\begin{aligned} m(\{P_1\}) &= p_1 - p_2 \\ m(\{P_1, P_2\}) &= p_2 - p_3 \\ &\vdots \\ m(\{P_1, \cdots, P_i\}) &= p_i - p_{i+1} \\ &\vdots \\ m(\theta) &= 1 - p_1 \end{aligned}$$

The final assignment above may not make much sense as it is shown. Note that the first *r* assignments proceed as discussed above. If we sum up the belief masses for the first *r* assignments, we find that the sum totals to p_1. Since we would like the belief function to total to 1.0, we have $1 - p_1$ mass left over, after the first *r* assignments. In the absence of any further information on how to distribute this belief mass, we assume that this quantity of belief mass represents our ignorance of the truth. This much mass is thus associated with the full universe of discourse, which, in our case, is the frame of discernment θ.

Belief functions that fit this template are called *Consonant Belief Functions* (CBF). [5] provides a look at a variety of different types of belief functions including the CBF. The CBF is ideally suited for applications where throughput cannot be expended liberally for the purposes of belief function construction. More rigorous methods, such as conditional embedding can be utilized to derive belief functions containing larger numbers of focal elements.

3.1.3 DEMPSTER'S RULE OF COMBINATION AND EVIDENTIAL CONFLICT

Fusion, in the Dempster-Shafer Calculus, occurs at the belief function level. Given two belief functions, m_1 and m_2, we can derive a fused belief function from these two by applying Dempster's Rule of Combination. Given two belief functions of the form

$$\begin{array}{ll} m_1(F_{11}) = m_{11} & m_2(F_{21}) = m_{21} \\ m_1(F_{12}) = m_{12} & m_2(F_{22}) = m_{22} \\ \vdots & \vdots \\ m_1(F_{1n}) = m_{1n} & m_2(F_{2m}) = m_{2m} \end{array}$$

Dempster's Rule creates a new fused belief function, where for every F_{1i}, and F_{2j}, a new focal element is formed of the form $m(F_{1i} \cap F_{2j}) = m_{1i} \times m_{2j}$. After all such products are computed, if a focal element appeared multiple times in the computations, all of the masses assigned to it are summed together and assigned to the focal element. We face an inherent risk, however, in performing such a computation. The risk arises from the fact that we may have $F_{1i} \cap F_{2j} = \varnothing$. But what does assignment of belief mass to the null set really mean? The null set only arises when there are no common elements between two focal elements being fused together. This means the null set only arises when we are trying to combine together two pieces of information that support different conclusions. That is, the null set indicates the presence of contradiction in the evidence being fused together. In other words, the total mass on the null set, in a belief function indicates the level of conflict in our evidence.

3.1.4 CONFLICT NORMALIZATION, ZADEH'S PARADOX AND SCORING ID CORRELATION

It is fairly easy to see that as we perform fusion via Dempster's Rule our belief mass on the conflict focal element can only grow. This leads to some unfortunate attempts at removing conflict from the picture altogether. The most frequently used technique for this purpose is conflict normalization. In brief, conflict normalization attempts to redistribute belief mass assigned to the null set, by dividing the mass on every focal element by the quantity $1 - m(\varnothing)$. In extreme cases, this can lead to the well-known Zadeh's Paradox. To illustrate this paradox, consider two belief functions to be fused together.

$$\begin{array}{ll} m_1(\{A\}) = 0.95 & m_2(\{B\}) = 0.95 \\ m_1(\{C\}) = 0.05 & m_2(\{C\}) = 0.05 \end{array}$$

Dempster's Rule gives us

$$\begin{array}{l} m(\varnothing) = 0.9975 \\ m(\{C\}) = 0.0025 \end{array}$$

If we normalize this belief function by dividing the focal elements by $1 - m(\varnothing) = 0.0025$, we get the following outrageous result.

$$m(\{C\}) = 1.0$$

Although very little of the evidence seems to support *C* as the correct conclusion, normalization leads us to conclude that the ID *must* be *C*! Instead of trying to remove conflict from a belief function, it may be wiser simply to accept it. Indeed, when we try to correlate attribute data against a tracked object, we often end up fusing the new attribute data with the attribute data that has cumulatively developed on the track. In doing such a fusion, we would like to gain some measure of the accuracy of correlation in ID space. The conflict associated with a fused result can aid us in determining whether we have correctly correlated the new attribute data to the track. The measure

$1-m(\varnothing)$ is a real-valued quantity bounded by the interval $[0,1]$. As the conflict in data being fused together increases, the measure moves toward a value of 0. The more consistent the evidence is, the closer the measure will be to 1.

3.1.5 PIGNISTIC MEASURES

One of the chief attributes of the Dempster-Shafer method that distinguishes it from a Bayesian method is the fact that belief mass is distributed across sets of candidate IDs, rather than on a specific ID. At some point in the process of tracking an object, we usually are faced with the dilemma of decisively declaring a single ID for the object. Unfortunately, given a belief function, it is not always easy to determine the best ID of the tracked object. The description of a tracked object's ID at the level of a belief function is referred to as a *Credal* measure of the ID. We need a measure at the decision level, rather than at the credal level. Such measures are called *Pignistic* measures.

The classical Pignistic measure is Pignistic Probability (see [4], for example). Given a belief function, the Pignistic Probability of a specific platform, say p_i is defined to be

$$\Pr(p_i) = \sum_{p_i \in F_j} \frac{m(F_j)}{|F_j|}$$

In the above, the F_j are the focal elements of the belief function, *m*, and $|F_j|$ denotes the cardinality of the focal element. The Pignistic Probability is bounded by two other pignistic measures, the support and plausibility measures.[1] The support for a platform p_i is defined as follows

$$\mathrm{Sup}(p_i) = \sum_{F_j = p_i} m(F_j)$$

The definition we have given above is a more specific instance of a general definition of support, which can be constructed for arbitrary sets of platforms. See, for example, [5]. The plausibility of a platform can be thought of as $1-\mathrm{Sup}(\overline{p_i})$, if we use the more general definition of support that is found in [5]. A more rigorous definition for plausibility is shown below.

$$\mathrm{Pl}(p_i) = \sum_{p_i \in F_j} m(F_j)$$

It is not difficult to see that for any given platform p_i, we have $\mathrm{Sup}(p_i) \le \Pr(p_i) \le \mathrm{Pl}(p_i)$. The support plausibility interval therefore, can be thought of as a definitive measure of how much a belief function supports a particular platform.

3.2 THE AD-HOC SCORING METHOD

Figure 1 below gives an overview of how this algorithm proceeds. Each of the sensor types we discussed in section 2 has associated with it some weight, just as it did in the Dempster-Shafer method. However, the domain of values for these weights is significantly different in this method. Let us assume that a report platform ID is similar in form to the platform confidence pairs introduced in section 2 ($\{(p_i, c_i) \mid p_i \in \mathbf{P}, 0 \le c_i \le 100\}$), with the exception that the confidence is replaced by a weight. This weight is dependent not only on the value of the

[1] There is some variation in the terminology here. Occasionally, what we refer to as a belief function is referred to as a Basic Probability Assignment and what we refer to as support is referred to as a belief function.

confidence associated with a particular platform, but also on the specific sensor that is reporting the evidence. A weight is not constrained by the interval [0,100] as a confidence is. Instead, a weight is essentially unbounded. A qualitative way to think about this concept is to tie the magnitude of the weight to the amount of trust we place in the sensor that is reporting the evidence. Suffice it to say that the evidence we are interested in fusing can be altered to match the form $\{(p_1, w_1), \ldots, (p_n, w_n)\}$, where the w_i represent weights.

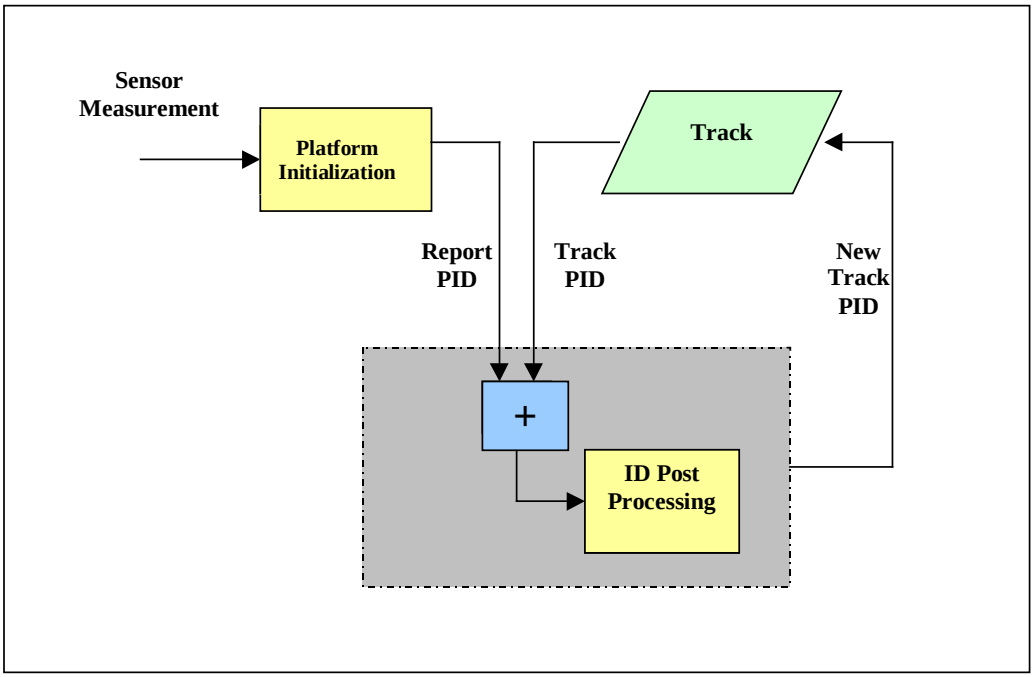

Figure 1: Overview of the Ad Hoc ID Fusion Algorithm.

Given two platform IDs, say $pid_1 = \{(p_1, w_{11}), \ldots, (p_n, w_{1n})\}$ and $pid_2 = \{(p_1, w_{21}), \ldots, (p_n, w_{2n})\}$, we can combine the evidence contained in these two platform IDs by simply adding together the evidential weights that correspond to the same platform. Hence, we may say that the fused ID result is given by $pid_f = pid_1 \oplus pid_2 = \{(p_i, w_i) \mid w_i = w_{1i} + w_2\}$. Given j distinct platform IDs we generalize this notation to $pid_f = \bigoplus_{k=1}^{j} pid_k = \left\{(p_i, w_i) \mid w_i = \sum_{k=1}^{j} w_{ki}\right\}$. Note that we assume that the cardinality of any platform ID is *n*. That is, any platform ID consists of exactly *n* ordered pairs, where we have a total of *n* platforms. We stress that some belief weights may have a value of 0.

What we have suggested so far is a very simple methodology for combining two platform IDs: we simply add up their weights. If there is some inconsistency inherent in the two pieces of evidence, we accept either interpretation. For example, if pid_1 places a lot of weight on platform p_x, say w_{1x} and pid_2 places weight w_{2x} on the same platform, where $w_{1x} >> w_{2x}$, the method of fusion we have discussed so far will include a large non-zero weight on p_x (specifically the weight will be $w_{1x} + w_{2x}$). Let us denote the fact that platform p_x is supported by a platform ID, pid_i by $p_x \in pid_i$, where we say $p_x \in pid_i$ iff w_{ix} is large. But what is large? Since this is a subjective question, whether $p_x \in pid_i$ or $p_x \notin pid_i$ is a fuzzy truth-value. But suppose we define $\delta(P)$ **as the degree to which the proposition *P* is true**. Then, we may write $\delta(p_x \in pid_1) > \delta(p_x \in pid_2) \Leftrightarrow w_{1x} > w_{2x}$. We further require that $w_{ix} = 0 \Leftrightarrow p_x \notin pid_i$.

Now, in terms of this notation, we can say that $p_x \notin pid_f = \bigoplus_{k=1}^{j} pid_k \Leftrightarrow \forall k, p_k \notin pid_k$. That is, as long as p_x is supported by one of the platform IDs being fused, it will be supported by the fused result. But, this is a very liberal definition of fusion. As long as a platform is declared a possibility by one of the platform IDs being fused, it remains a possibility in the fused result. This is borne out by the fact that for any *k*, $\delta(p_x \in pid_f) \geq \delta(p_x \in pid_k)$.

We can now take the opposite tack in constructing another fusion algorithm. Let us say that given two platform IDs, a platform is supported in the fused platform ID iff it is supported in both constituent platform IDs. So, given two platform IDs pid_1 and pid_2, we define $pid_f = pid_1 \otimes pid_2 = \{(p_i, w_i) | w_i = w_{1i} \circ w_2\}$, where

$$w_{1i} \circ w_{2i} = \begin{cases} w_{1i} + w_{2i} & w_{1i}, w_{2i} \neq 0 \\ 0 & Otherwise \end{cases}$$

As with the previous algorithm, we generalize our notation for fusion of *j* platform IDs as follows.

$$pid_f = \bigotimes_{k=1}^{j} pid_k = \left\{ (p_i, w_i) | w_i = \bigcirc_{k=1}^{j} w_{ki} \right\}$$

Under this definition of ID fusion, we have $p_x \in pid_f = \bigotimes_{k=1}^{j} pid_k \Leftrightarrow \forall k, p_k \in pid_k$. That is, in order for the fused result to support a platform p_x, every platform ID that is being fused must support p_x. It is readily apparent that this definition of fusion is essentially complementary to the first ID fusion operation and that it is a much more restrictive form of fusion. A platform is part of the fused result iff it is a part of every constituent piece.

3.2.1 THE KNOWLEDGE POINT AND THE PRUNING THRESHOLD

We have given two examples of how platform IDs can be fused together to produce a unified result. The two methods we have illustrated represent the two extremes of potential approaches to the problem. In the first method, we were very permissive, allowing all potential conclusions supported by any constituent piece of evidence to be supported by the fused result. In the second method, we were very restrictive, requiring that the potential solutions supported by the fused platform ID be supported by every constituent piece of evidence. We now attempt to find the middle ground of the process.

To do so, we realize that the process of accumulating ID evidence is a continuous one in more than one sense. Our level of ignorance of the true ID of the observed target, if such a measure can be quantified in terms of percentages, begins at 100%. At each step, accrual of new ID evidence drives this percentage down. In the ideal situation, we would like our ignorance to be identically zero at the end of the process. In reality, we can never be sure whether true ID matches our computed ID. There is always a chance, albeit small, that we may have derived the wrong conclusion on the ID of the observed target. Hence, ignorance asymptotically approaches zero, as we accrue evidence, in the best case. In worse cases, if platform IDs remain diffuse, the ignorance level of the ID of the observed object will asymptotically approach some non-zero value. For example, if the only ID we receive is one derived from a CNI source (notoriously ambiguous) that is ambiguous between a large number of platforms, our ignorance level will not asymptotically approach zero. Instead, we shall always be ignorant between the platforms this single sensor type returns. In the worst case, obviously, our ignorance remains at 100%. This case arises if we have a sensor that is malfunctioning or performing very poorly (and thus not being able to eliminate any platform in the universe of discourse). The graph in Figure 2 below illustrates this point.

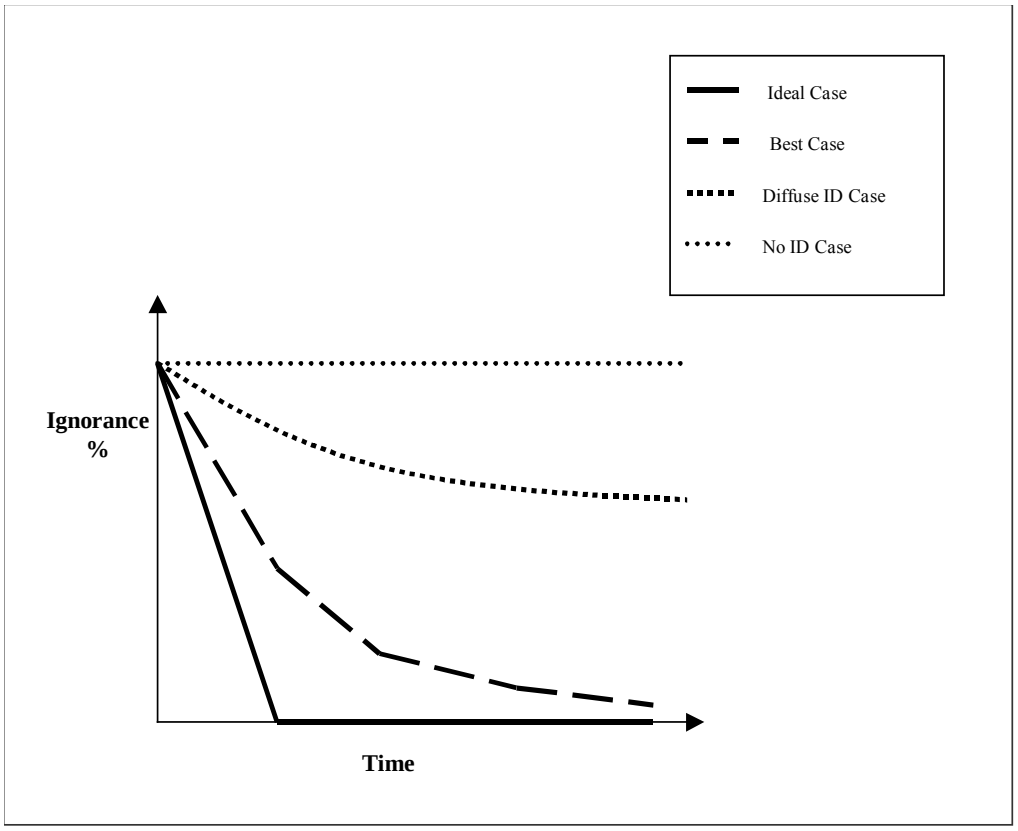

Figure 2: Plot of Ignorance vs. Time for Various Cases.

Just as our level of ignorance is a continuous variable that hopefully drops over time, our willingness to allow alternative hypotheses on the true ID of the observed object should be a continuous variable. At the beginning of the process, since we have very little evidence to support any one conclusion, we may be very permissive in allowing multiple potential conclusions to be part of the fused solution. However, as we accrue evidence, our ability to discern an incorrect hypothesis grows, and we become less and less permissive about which platforms should be allowed to be part of the fused ID.

This line of reasoning suggests that we use a fusion algorithm that is a hybrid of the two approaches that we have discussed so far. Let us consider again the structure of a platform ID, $pid = \{(p_1, w_1), \ldots, (p_n, w_n)\}$, associated with a tracked object. If we consider *pid* to be a collection of hypotheses about the true ID of the object that is being tracked, then any given p_x is associated with the proposition that the true ID of the object being tracked is the platform p_x. Our belief in this proposition is represented by the belief weight w_x. While $\delta(p_x \in pid) = w_x$ for all p_x is relatively low, we say that *pid* is in an open world assumption. This is equivalent to saying that we will not discourage or deny the introduction of alternative hypotheses into *pid* while $\delta(p_x \in pid)$ is low for all p_x.

Let us define the *Knowledge Point* as the point in time when we disallow the introduction of new propositions into *pid*. Let us say that we associate some belief weight W_{Kpt} with the Knowledge Point. In order for us to disallow the introduction of new propositions into *pid* (this process of disallowing new propositions is called the closed world assumption and ***pid* is said to be mature when this point is reached**), we must have at least one *x* such that $\delta(p_x \in pid) \geq W_{Kpt}$. Once we meet this condition, we have made the fundamental assumption that the true ID of the tracked object is one of the platforms that is contained in the platform ID *pid.* Thus, our task now becomes to eliminate as many of the propositions from *pid* as possible.

Let $\max(\{w_1, \cdots, w_n\}) = w_x$. Since we are assuming that we are in the closed world assumption, we have $w_x \geq W_{Kpt}$. Let us say that if pid supports a candidate platform with a weight that is within some ΔW of w_x, we retain the candidate platform. Otherwise we prune away the candidate platform from contention. Pruning away a platform entails dropping its belief weight to 0. We refer to ΔW as the pruning weight. The process of pruning is shown in the figure below. As the diagram illustrates, $A, E, F \in pid$, but $A, E, F \notin \text{Prune}(pid)$.

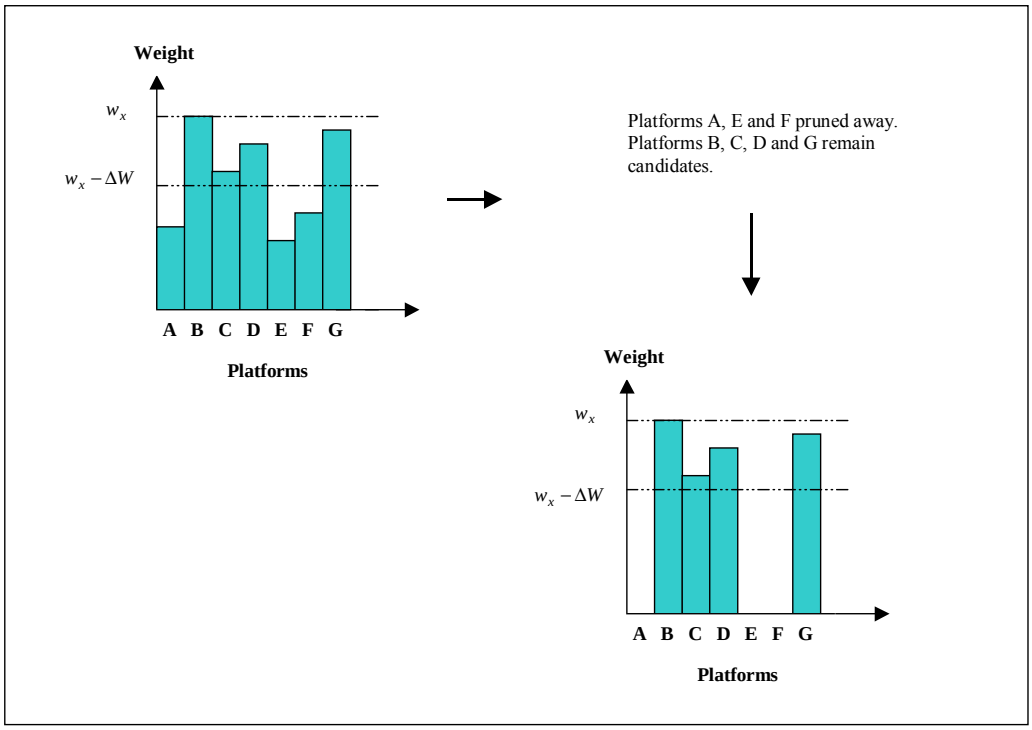

Figure 3: Pruning a Platform ID that has Reached the Closed World.

3.2.2 SCORING THE COMPATIBILITY OF TWO PIDS

While maintaining a fused ID on a target of interest, we would like in general to fuse platform IDs that are consistent with each other. That is, if, on a track, we have a cumulative fused ID indicating a preference for platform p_x, and we receive evidence from a sensor source that indicates a preference for a platform other than p_x (say p_y), we should be able to detect that the two platform IDs are incompatible and thus, probably, should not be fused together.

One way to construct a metric that captures this type of contradiction in the evidence is to consider the two extreme types of fusion algorithms we have discussed above. Before we discuss how these two algorithms might help, we state here that for all platforms p_i, $\delta(p_i \in pid_1 \otimes pid_2) \leq \delta(p_i \in pid_1 \oplus pid_2)$. Given platform IDs pid_1 and pid_2, if we take $\max(pid_1 \oplus pid_2) = w_1$ and $\max(pid_1 \otimes pid_2) = w_2$, for pruning purposes, we have two separate points from which to view the pruning test. But given the ordering on these two fused results, we conclude that $w_2 \leq w_1$. If the platform with the most support in $pid_1 \otimes pid_2$ would be pruned out in $pid_1 \oplus pid_2$, we can say that the two platform IDs are incompatible. That is, if $w_2 \leq w_1 - \Delta W$ (i.e. the platform corresponding to w_2 would have been pruned out of $pid_1 \oplus pid_2$), then we can say that the two platform IDs are incompatible.

Simplifying $w_2 \leq w_1 - \Delta W$, we have that the two platform IDs are incompatible iff $\frac{w_1 - w_2}{\Delta W} \geq 1.0$. If we impose the restriction that our metric must return 1.0 when the two platform IDs are a good fit and some number less than 1.0 if they are not, we can rewrite the above as $1 - \frac{w_1 - w_2}{\Delta W} \leq 0$ if the two platforms are incompatible. Otherwise, if the two platforms are compatible to some degree, we have $0 < 1 - \frac{w_1 - w_2}{\Delta W} \leq 1$. The degree to which we require the platform IDs to be compatible is a subjective matter, dependent on the application to which this metric is being applied. Intuitively, it is easy to see that the two platform IDs are perfectly compatible iff $w_1 = w_2$.

It is also fairly easy to see that the metric has a negative value if the maximally supported platform in $pid_1 \otimes pid_2$ would have been pruned away in $pid_1 \oplus pid_2$. The metric is identically 0 if the maximally supported platform in $pid_1 \otimes pid_2$ is exactly ΔW less than the maximally supported platform in $pid_1 \oplus pid_2$. The metric is positive if the maximally supported platform in $pid_1 \otimes pid_2$ would not have been pruned away in $pid_1 \oplus pid_2$.

3.2.3 PULLING TOGETHER THE COMPLETE ALGORITHM

So far, we have described the various pieces of the variant scoring based fusion algorithm. In this section, we will bring together all of the pieces of the algorithm that we have discussed.

Given two platform IDs pid_1 and pid_2, that we want to fuse together, we begin by calculating the two extreme fusion values $pid_1 \otimes pid_2$ and $pid_1 \oplus pid_2$. If neither of the platform IDs is mature, we calculate the compatibility score of $pid_1 \oplus pid_2$ and return this liberal fusion result along with the compatibility score as the output of the fusion process. The rationale again, is that we want to be as permissive as possible, initially, to allow as many potential platforms as possible to be included in the candidate list. Once maturity is reached, we can prune away unlikely candidates.

If either of the platform IDs is mature, we see if there are any platforms that are supported by $pid_1 \otimes pid_2$. If no such platforms exist, we can assume that the two IDs are incompatible and return a compatibility score of 0. Otherwise, we have a fused result that is in the closed world. We prune the platform ID $pid_1 \oplus pid_2$, using the weight of the maximally supported platform in the platform ID $pid_1 \otimes pid_2$. If pid_1 and pid_2 are compatible, it stands to reason that the maximally supported platform in $pid_1 \oplus pid_2$ and $pid_1 \otimes pid_2$ are identical. As a result, those platforms with a low degree of support in $pid_1 \oplus pid_2$ will be pruned away.

On the other hand, if we have the fusion of two incompatible platform IDs, the maximally supported platform in $pid_1 \otimes pid_2$ (say p_1) will be different from the maximally supported platform in $pid_1 \oplus pid_2$ (say p_2). Further, we can say that $\delta(p_1 \in pid_1 \otimes pid_2) < \delta(p_2 \in pid_1 \oplus pid_2)$, and thus, if we prune by the weight on p_2, we run the risk of pruning out the platforms that are in common between pid_1 and pid_2. Therefore, we prune using the weight on p_1 instead. We let the compatibility score drive the decision on whether to accept or reject the fusion of these two platform IDs. The compatibility score and the pruned result are returned as the output of the fusion.

An example of how fusion of a mature platform ID and an immature platform ID proceeds is shown in the figure below. In this figure, we show the fusion of a platform ID that has been developing on a track, and a platform ID derived from a sensor report.

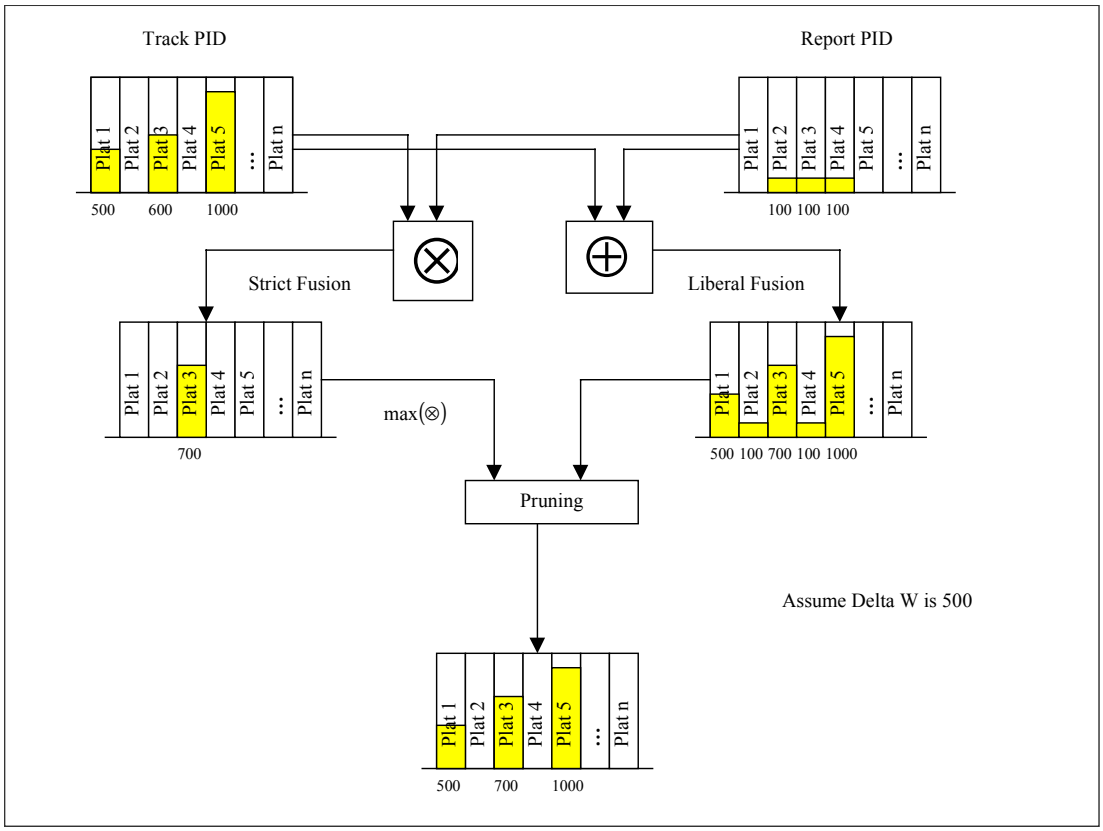

Figure 4: An Example of the Complete Ad Hoc Fusion Algorithm.

4.0 TESTING METHODOLOGY AND SCENARIO DESCRIPTION

In this section we will describe our analysis tools and environment, as well as the specific scenario on which we are demonstrating the performance of the two algorithms discussed in this paper.

We tested the Dempster-Shafer and Ad Hoc methods by embedding these algorithms in a larger simulation of an actual fighter program's sensor fusion system. This larger simulation, coded in Mathematica, is hosted on both PC and VAX workstations. Specifically, we were interested in the performance of these algorithms in the context of a pre-scripted scenario that was fairly dense with friend and foe platforms. The scenario we ran against is summarized in the chart below.

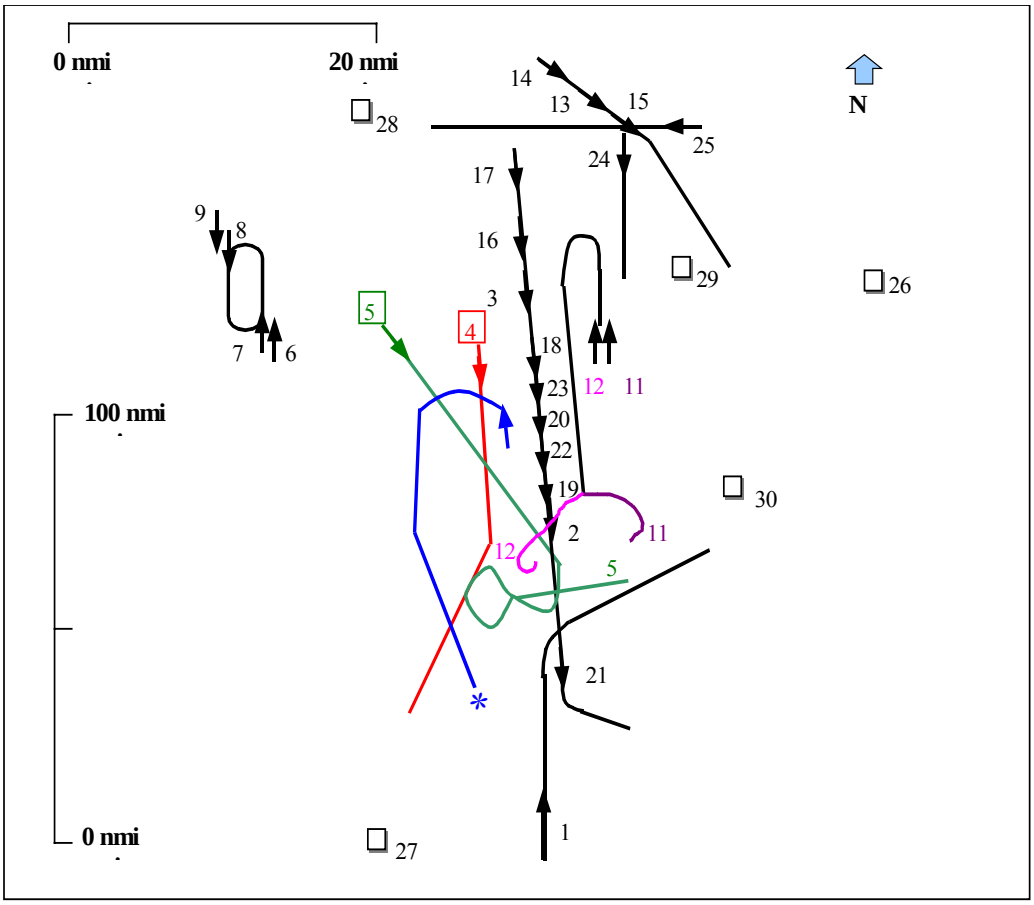

Figure 5: A Graphical Description of Target Paths in the Test Scenario.

In this figure, ownship is marked with the number 1. The remaining air targets in the scenario are marked with numbers from 2 to 25. Ground targets are marked with boxes, and enumerated from 26 to 30. Air targets are denoted by arrows that indicate their heading. There are three major formations in this scenario that are of interest. Four enemy fighters move in a cap formation northwest of ownship and are enumerated from 6 to 9. A long corridor of disparate friendly platforms (numbered 2, 19, 22, 20, etc.) head toward ownship. A third much smaller corridor of mixed friendly/foe platforms maneuver to the northeast of ownship (13, 14, 15, 24, 25). In this paper, we will only analyze the performance of our algorithms on the large corridor of friendly platforms heading towards ownship. A study of this feature of the scenario is instructive, because of the kinematic characteristics of this corridor. We elaborate on this below. Such conditions make it possible to somewhat isolate the performance of ID on this set of targets.

It is also significant to mention that the results shown in the subsequent section assume that the Dempster-Shafer algorithm rejects all reports that do not correlate with a metric score of 0.85 or greater. The ad hoc method is a lot more liberal, allowing anything with a non-zero metric score value to update a fused result.

Broadly speaking, there are two ways to judge the performance of an algorithm; how quickly does the algorithm execute and how "good" is the result of the algorithm in providing correct ID. The first of these questions is fairly easily answered. Timing tests on the two algorithms, running on the PC, indicated that both completed executing the described scenario in about the same amount of time. Below we are going to be discussing the results we received on one particular, fairly typical run. In the run we discuss, the Dempster-Shafer approach took slightly less time (~91.3% of the time it took to run the ad hoc method). However, in other circumstances, the Ad Hoc method beat out the Dempster-Shafer run by similar figures. It is, therefore, difficult to declare one algorithm faster than the other. Factors such as other processes running on the machine, platform type (i.e. PC or DEC) and medium of program (i.e. Mathematica) may contribute to variability in the speed performance of these algorithms.

The other side of the question is how well did the two algorithms do in determining ID? We show performance of our two algorithms, in this regard, on system tracks created during the execution of the scenario via graphs of the type shown in Figure 6 below. This graph shows that our mission database consisted of platforms enumerated contiguously from 1 to 55. These are shown on the y-axis. Along the x-axis, we show the scenario time. At a given time, a point on the chart indicates the presence of the corresponding platform in the platform ID for the track associated with the graph. The strength of the point indicates the degree to which the platform ID supports the platform (i.e. does the platform have a lot of weight or belief mass, or does it have very little?). The white space between platform ID points indicates that no new evidence in the form of sensor attribute data has been received to change the platform ID. For example, in the figure below, no new evidence has entered the system between time unit 800 and time unit 1125. Therefore, during this period of time, we can conclude that there has been no change in the platform ID on this track. The header of the figure indicates that we are looking at performance for a system track on target 2, as shown above in the scenario description, and that the system has allocated the number 25 for this track.

Given a stream of attribute data, if we make the assumption that we absolutely believe every single piece of evidence, we can define the "True ID" value as the intersection of the sets of platforms each piece of evidence supports. Returning to our definition of a platform ID supporting a platform described in section 3.2 ($p_i \in PID \Leftrightarrow w_i > 0$), given a stream of platform IDs, let the True ID set be the set of platforms that is supported by every one of the platform IDs (given n Platform IDs, the True ID set is $\{p_i \mid p_i \in PID_k, \forall k = 1, \cdots, n\}$). Those platforms that make up the true ID set are shown in the graph using the shaded boxes. As we would expect, over time, the cardinality of this set decreases. In the graph shown, the cardinality actually drops down to 1 around time unit 110.

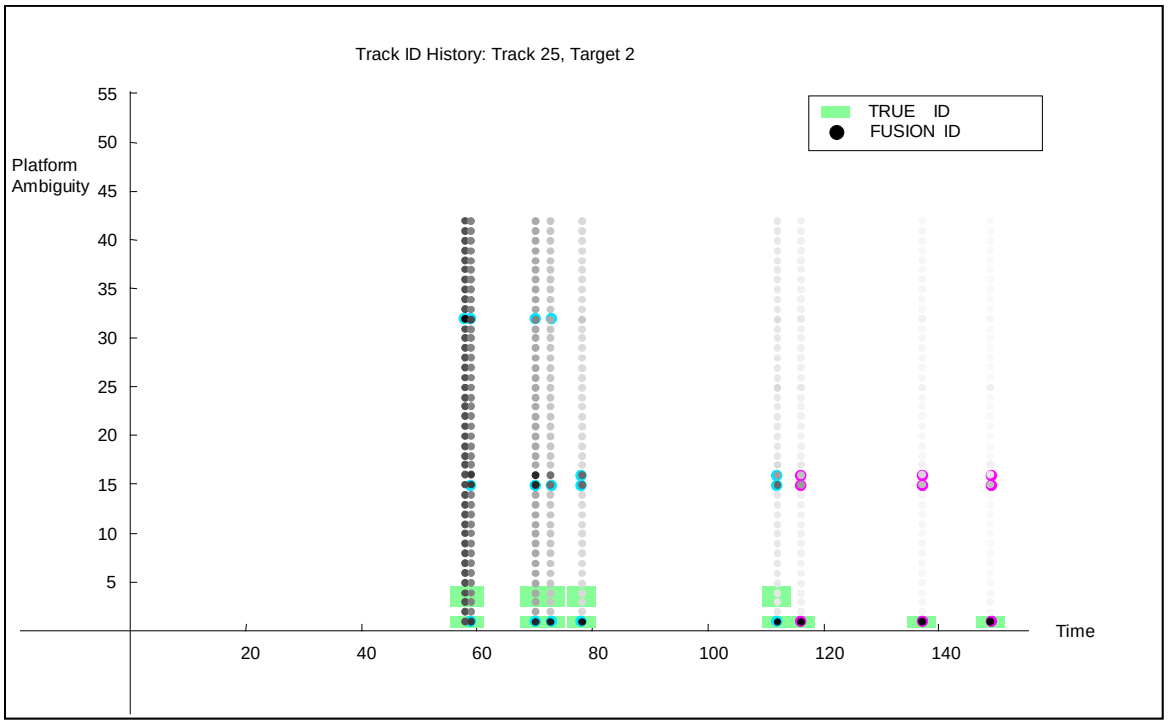

Figure 6: ID Performance of Track 25 (Target 2) in Dempster-Shafer Run.

5.0 ANALYSIS RESULTS

Figure 6 above shows performance of the Dempster-Shafer algorithm in tracking target 2. As the graph illustrates, by time unit 120, the ID on the track has been narrowed significantly. In fact, the only platform with any significant weight is the one that corresponds to the true ID. With ownship heading of 0^o, Target 2, as our scenario

description indicates, is part of the long train of targets with heading 180° (i.e. they are on a path directed towards ownship). Given the relative kinematic ambiguity in angle measurements along the line of sight connecting ownship with the train of targets, we have to rely on ID scoring to differentiate the various targets. The Ad Hoc method displays a deficiency in this regard. Whereas the Dempster-Shafer method is able to differentiate the target on the basis of ID, and create a separate track within 60 time units, the Ad Hoc method is only able to pick up the target much later in the scenario.

The ID platform ambiguity of the track it forms on the target is shown below in Figure 7. As we see, this track does not form until about 700 time units into the scenario. Further, it is not able to gather enough information to narrow down the ID to a single platform. It makes some progress towards this goal between time unit 1000 and time unit 1200. However, due to a loss of information around time unit 1200, it is unable to sustain this movement. By the end of the run, it returns a platform ID list that is ambiguous between a large number of platforms. But this cannot be considered a huge advantage for the Dempster-Shafer method. The reason is that a similar track also forms for this method around the same time. Whether this new track exists concurrently with track 25 above, or whether track 25 is dropped (this can happen because of the lack of any new data over a period of time) causing a new track on target 2 to form at a later time, the same problem arises in both methods.

The difficulties of using the scoring mechanism of the Ad Hoc method to differentiate targets on the basis of ID alone are illustrated again by the tracking of target 3. While the Ad Hoc method was able to eventually create a track on target 2, it failed altogether in tracking target 3. The Dempster-Shafer method, on the other hand, worked quite well, as illustrated in Figure 8. Within 450 time units, the Dempster-Shafer method had correctly identified the target. By time unit 650, the ambiguity of the ID drops to 1.

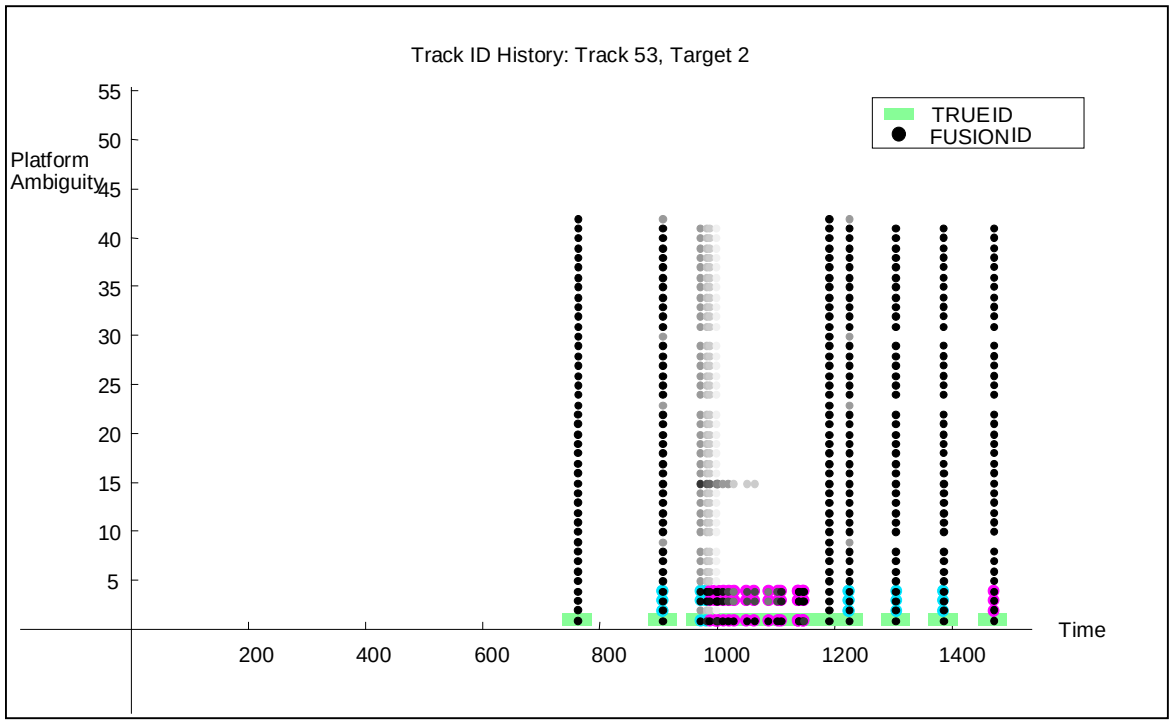

Figure 7: ID Performance of Track 53 (Target 2) in Ad Hoc Scoring Run.

The permissive nature of the scoring in the Ad Hoc method leads not only to problems with targets going undetected, but also with miscorrelation of data. Consider Figure 9 below which shows the performance of the Ad Hoc scoring method on target 16. Because of the kinematic ambiguity of the situation under consideration, sensor reports gate incorrectly to tracks using just kinematics. Since the scoring is very permissive in ID as well, any track that gates with a sensor report in kinematics, and has even one platform in common with the report's platform ID, will successfully attract the report. The tracking on target 16 shows this difficulty clearly. Reports that do not truly belong to a target track can hijack the track easily, as this case illustrates. By 180 time units, the ID on the track

does not reflect the True ID of the target being tracked. This problem, quite clearly, does not arise in the context of a Dempster-Shafer run, as the track on the same target using Dempster-Shafer clearly indicates in Figure 10.

Given these results, it would seem that on this scenario run the Dempster-Shafer method seemed to perform somewhat better than the Ad Hoc method. However, it would be imprudent to conclude definitively, at this stage of analysis that the Dempster-Shafer method is the better overall algorithm. Further study and analysis needs to be performed to reach any such definitive conclusion.

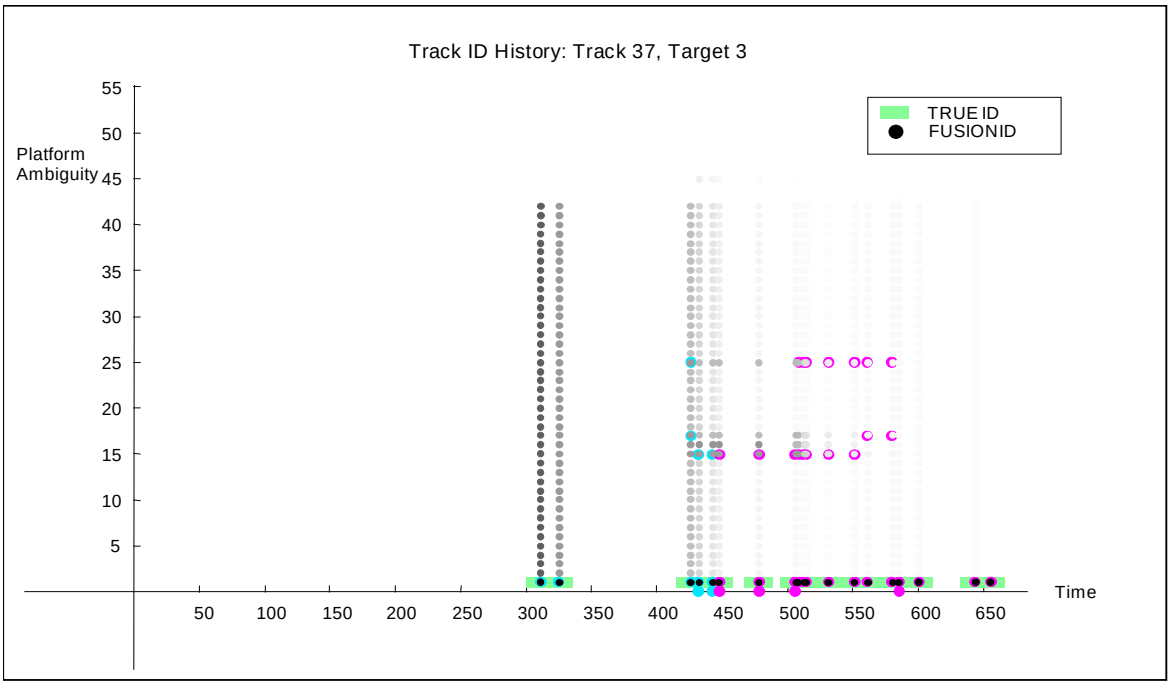

Figure 8: ID Performance of Track 37 (Target 3) in the Dempster-Shafer Run.

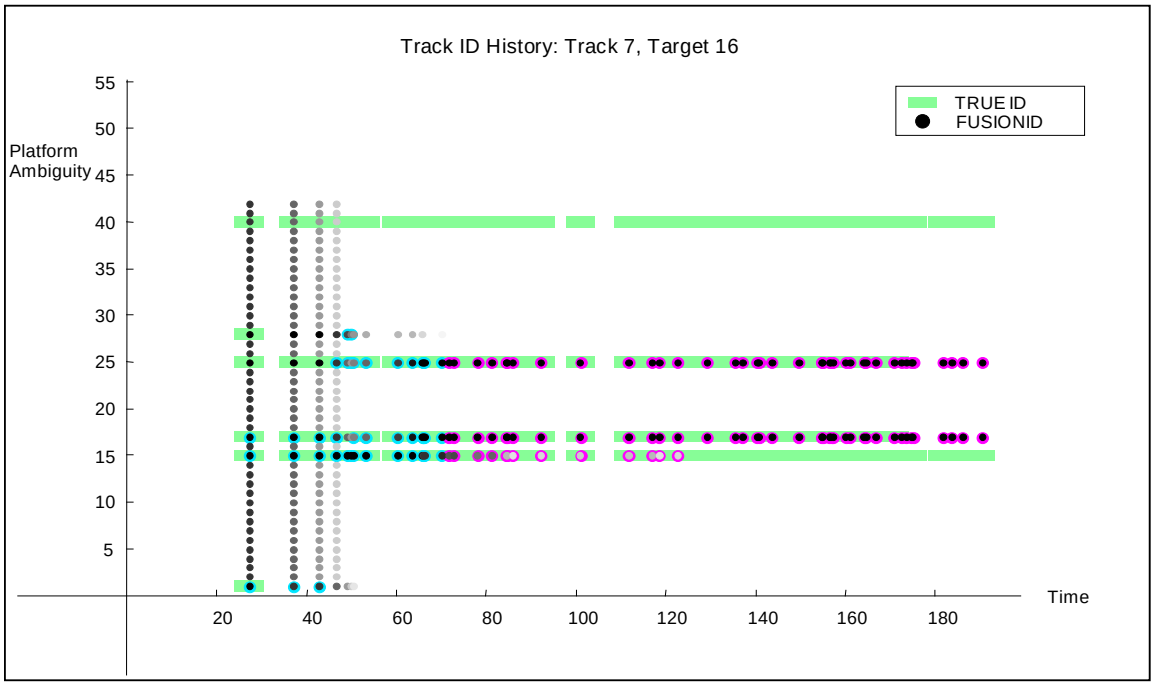

Figure 9: ID Performance of Track 7 (Target 16) in the Ad Hoc Scoring Run.

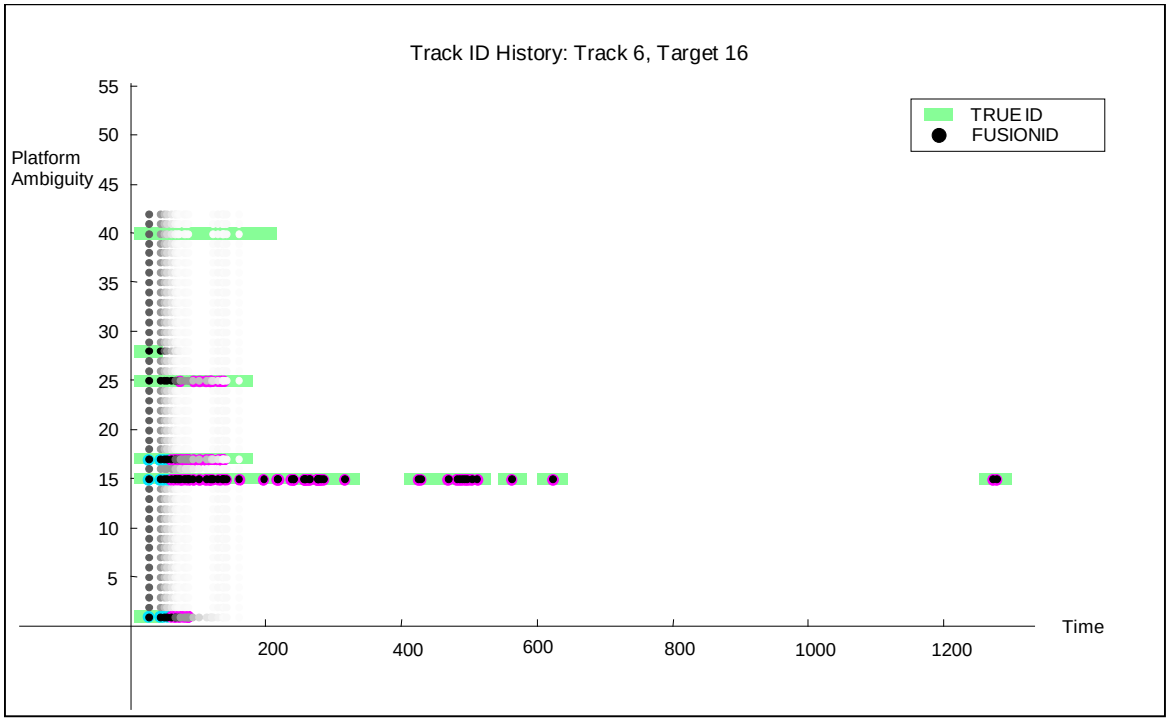

Figure 10: ID Performance of Track 6 (Target 16) in the Dempster-Shafer Run.

6.0 SUMMARY

Given the interim nature of the analysis presented in this paper, we have pointed out the areas in which the Dempster-Shafer algorithm seems to work better than the present Ad Hoc scoring method being employed on the present algorithm. In all fairness, we should note that the Ad Hoc scoring method does work just as well on slightly simpler scenarios. Even in the context of the scenario discussed in this paper, the performance on the cluster of targets formed by targets 6 through 9 is almost identical in the Dempster-Shafer and the Ad Hoc Scoring runs. There are other aspects to consider as well, which may contribute to better performance for the Dempster-Shafer method. We are using a fairly restrictive scoring metric in the case of Dempster-Shafer, whereas the scoring metric for the Ad Hoc method is highly permissive. Since the Ad Hoc method is more commonly used on high performance fighters, we have not changed the accept/reject value of the metric already in the simulation. The performance of the Ad Hoc method may improve if we change this accept/reject value to something that is less permissive. A thorough analysis on the optimal value for the scoring metric for both methods should be performed before a final judgment is made on the best algorithm for the application.

ACKNOWLEDGEMENTS

We would like to acknowledge Michael Smith and Gary Fogle of the Air Force Research Laboratory at Wright Patterson Air Force Base, for encouraging the exploration of alternative fusion algorithms. We would also like to express our gratitude to the members of the Boeing Multi-Sensor Fusion Analysis and Sensor Track Fusion groups for their encouragement and support for the completion of this work.

REFERENCES

1. F. W. Cathey and M. Munoz. Proc. IRIS Sensor & Data Fusion, 1998. Vol. 1, pp. 115-128.

2. R. Loo. *Identification Algorithms.* ID Appendix to Program SDAR, 1999.

3. G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, 1976.

4. P. Smets. *Constructing the Pignistic Probability Function in a Context of Uncertainty*. Uncertainty in Artificial Intelligence 5 (Eds. M. Henrion, et al). Elsevier, 1990. pp. 29-39.

5. R. R. Yager. *Entropy and Specificity in a Mathematical Theory of Evidence.* Int. J. General Systems, 1983. Vol. 9, pp. 249-260.